Evolution in Electrical and Electronic Engineering Vol. 2 No. 2 (2021) 308-317

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Hardware Implementation of a Cryptographic Coprocessor

Daniel Tan Boon Kai[1], Chessda Uttraphan A/L Eh Kan[1,2]*

[1]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

[2]VLSI and Embedded System Technology (VEST) Focus Group, FKEE,
Universiti Tun Hussein Onn, Batu Pahat, 86400, Johor, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2021.02.02.037
Received 22 July 2021; Accepted 15 September 2021; Available online 30 October 2021

Abstract: The purpose of this project is to design the hardware of the cryptographic coprocessor by using Verilog Hardware Description Language (HDL). Cryptographic coprocessor has been widely used to offload the compute-intensive cryptography tasks from the main processor due to its performance and efficiency compared to the pure software solution. The proposed coprocessor comprises an input register, 16x16 register files, instruction decoder, control unit, arithmetic logic unit (ALU) and the hashing unit. The hashing unit is implemented using a non-linear look-up table (LUT). The instruction set architecture (ISA) consists of 12 instructions that execute independently from the main processor. The 16-bit hash circuit provides a faster way of implementing a hash function for any cryptographic algorithms as compared to software implementation. The design is simulated to verify the functionality of the proposed ISA using the ModelSim Intel Field Programmable Gate-Array (FPGA). Simulation results show that the proposed cryptographic coprocessor produces the correct outputs as in specifications.

Keywords: Cryptographic Coprocessor, Verilog HDL, Hash Function

1. Introduction

Nowadays, technologies and internet system have become one of the most important and common things in this human daily life. Networks is the items that any company had to achieve competitive advantage. However, all communication or message data over any network, mostly the networks needs to be secured for preventing the intruder from breaking through and get any confidential and sensitive data [1]-[3]. Network not only used by company but also used by home users. This bring in cryptography as major component to secure the communication over network [4]-[6]. In the last few years, the importance of cryptography related to security of electronic data transactions has acquired critical significance. Many users had interchanged and generated large amount information in several field through telephone conversations, e-commerce transactions and also internet. This can be earned

*Corresponding author: chessda@uthm.edu.my

by several techniques such as cryptography, biometric and password. In this perception, cryptography techniques are mainly useful.

This project presents a design of a cryptographic coprocessor on FPGA. The hardware implementation of the cryptographic coprocessor is faster and less prone to exploitation than the software implementations. The proposed system comprises an input register, 16x16 register files and combinational logic (the core design of the coprocessor). The functionality and performance of the proposed cryptographic coprocessor will be verified through a series of simulations.

2. Materials and Methods

This section presents the design of the cryptographic coprocessors in which consists of two main parts as follows: The first part describes the block diagram of the proposed design. The second part explains the software specification for simulation.

The proposed system comprises an input register, 16x16 register files, and combinational logic (the core design of the coprocessor). The clock signals know how to execute the programmed functions, and the reset will set the value again to 0 and 1. Figure 1 illustrates the block diagram of the coprocessor.

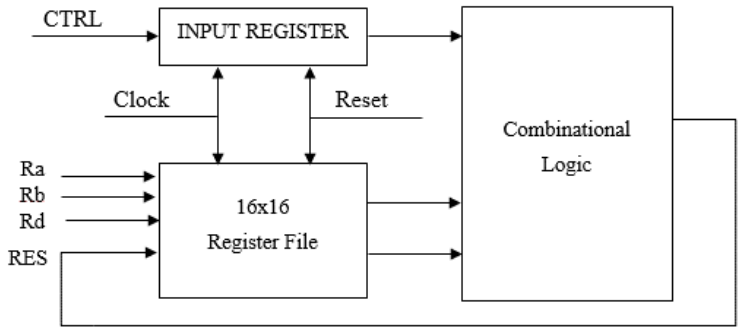

Figure 1: Block diagram of coprocessor

The Combinational Logic shown in Figure 1 consists of three major modules of the combinational logic unit such as ALU, Shifting unit, and Nonlinear Lookup Operation. Figure 2 shows the block diagram of the combinational logic unit.

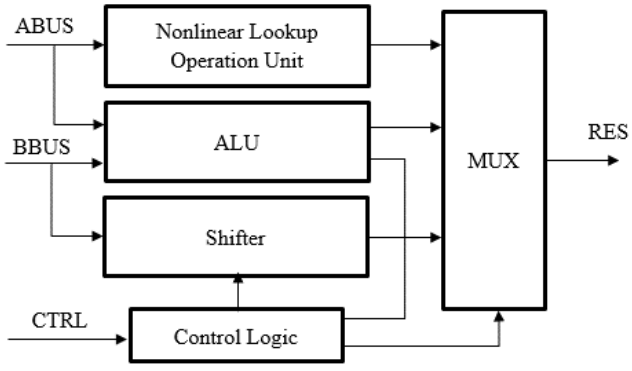

Figure 2: Block Diagram of Combinational Logic Unit

2.1 Register File

Registers are the temporary storage locations inside the CPU that hold the addresses and data. The register file is a component that contains multi-purpose registers of the microprocessor. 16x16 register file means 16-bit value data which operate with the 16 memory location. Ra, Rb and Rc represent the signal data input and dual-line for the signal, which is against combinational logic. Figure 3 shows the register file design.

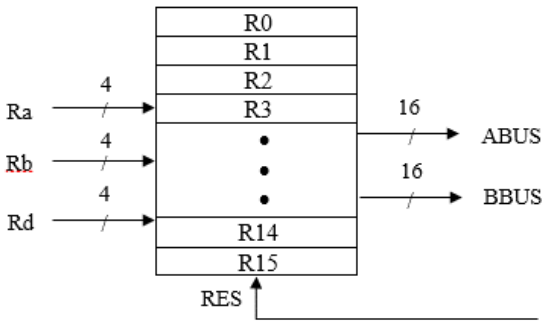

Figure 3: Register File

2.2 ALU (Arithmetic Logic Unit)

Figure 4 shows the block diagram consists of the ALU, input and output. ALU is a part of the coprocessor that use to carry out the logic operations and instruction set architecture of the coprocessor. The operation of ALU is shown in Table 1.

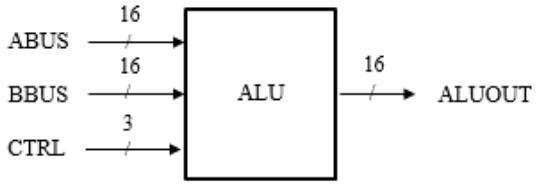

Figure 4: Block Diagram of ALU

Table 1: ALU Operation

Code (binary)	Operation
0000	ADD : ALUOUT ← ABUS + BBUS
0001	SUB : ALUOUT ← ABUS - BBUS
0010	AND : ALUOUT ← ABUS & BBUS
0011	OR : ALUOUT ← ABUS | BBUS
0100	XOR : ALUOUT ← ABUS ^ BBUS
0101	NOT : ALUOUT ← ~ABUS
0110	MOV : ALUOUT ← ABUS
0111	NOP : No Operation

2.3 Shifter Unit

A shifter unit is the main component in the prospective processing unit of the coprocessor. A shifter can shift and rotate data mostly used in the transpositions of ciphers and permutation. Fast rotating and shifting functions are essential for cryptographic applications. The simple block diagram of the Shifter is shown in Figure 5, and Table 2 shows the operation of the Shifter.

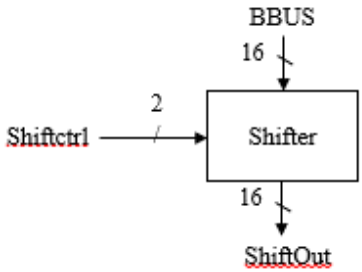

Figure 5: Block Diagram of Shifter

Table 2: Shifter Operation

Code	Operation
1000	ROR8 :Rotate right by 8 bit
1001	ROR4 :Rotate right by 4 bit
1010	SLL4 :Shift left logical by 8 bit

2.4 Nonlinear Lookup Table (LUT)

The block diagram of the Combination Logic from Figure 2 consists of the nonlinear lookup table operation unit. Figure 6 shows the simple block diagram for the Nonlinear Lookup Operation with the 5-bit value for S-Box 1 and S-Box 2:

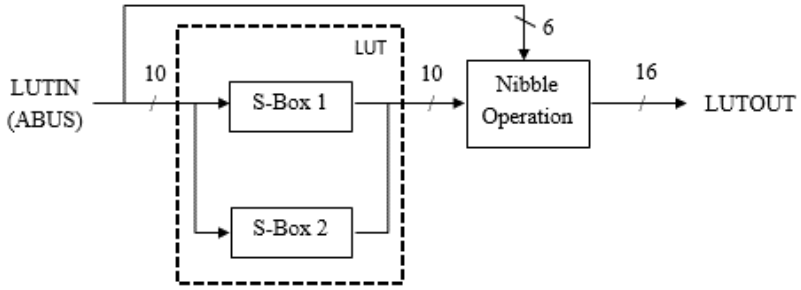

Figure 6: Block Diagram of Nonlinear Lookup Operation

2.5 Control Logic (CTRL)

The block diagram of the control logic is shown in Figure 7 and simple logic to apply the control logic as shown in connection with the truth table summarize in Table 3.

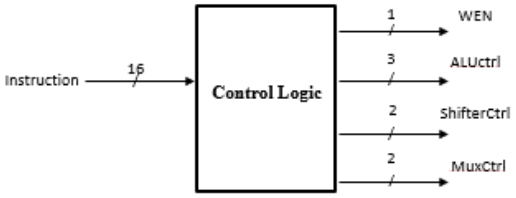

Figure 7: Block diagram of cryptographic coprocessor control unit

Table 3: Truth table for the operation control bits (opcode)

Instruction		Outputs: Control Signals			
Operation	Opcode	WEN	ALUctrl	ShifterCtrl	MuxCtrl
ADD	0000	1	001	00	01
SUB	0001	1	010	00	01
AND	0010	1	011	00	01
OR	0011	1	100	00	01
XOR	0100	1	101	00	01
NOT	0101	1	110	00	01
MOV	0110	1	111	00	01
NOP	0111	1	000	00	01
ROR8	1000	1	000	01	10
ROR4	1001	1	000	10	10
SLL8	1010	1	000	11	10
LUT	1011	1	000	00	11

2.6 Instruction Set Architecture

Instruction set architecture (ISA) outlines which instructions can be carried out by the computer. Set of Assembly language memories represents the code of the particular computers. The instruction into a machine has depended on the computer that is different processors have different instruction sets. The Table 4 shows the details of the Instruction Set Architecture will implement for the project and Figure 8 illustrates the operation for a cryptographic coprocessor when the control logic receives the command or instruction from the user and operates it one by one to prevent both commands from coming simultaneously.

Table 4: List of Instruction

Instruction	Operation
0000 xxxx xxxx xxxx	ADD : Rd ← Ra + Rb
0001 xxxx xxxx xxxx	SUB : Rd ← Ra - Rb
0010 xxxx xxxx xxxx	AND : Rd ← Ra & Rb
0011 xxxx xxxx xxxx	OR : Rd ← Ra | Rb
0100 xxxx xxxx xxxx	XOR : Rd ← Ra ^ Rb
0101 xxxx xxxx xxxx	NOT : Rd ← ~Ra
0110 xxxx xxxx xxxx	MOV : Rd ← Ra
0111 xxxx xxxx xxxx	NOP : No Operation
1000 xxxx xxxx xxxx	SR : Rd ← Rb >> 2 (8bit)
1001 xxxx xxxx xxxx	SR : Rd ← Rb >> 1 (4bit)
1010 xxxx xxxx xxxx	SL : Rd ← Rb << 2
1011 xxxx xxxx xxxx	LUT : Rd ← Ra[LUT]

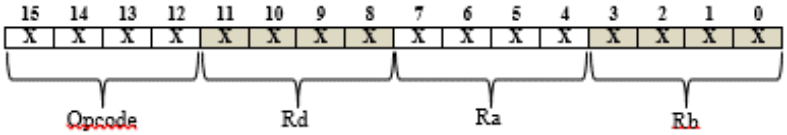

Figure 8: 4 instruction fields with the number size of 4-bits

2.7 Methods

The design of cryptographic coprocessors has been coded in form of Verilog code. The result for each processing unit of cryptographic coprocessor in FPGA is simulated and synthesized based on RTL simulation from Quartus Altera II. All results are shown in figures and the design has been simulated by writing a Testbench.

3. Results and Discussion

3.1 Results

Figure 9 to Figure 17 show the ISA simulation results obtained from the Coprocessor. As observed in the Figure 9, this figure shows the ISA simulation result of the coprocessor that presents the twelve ISA operation which shown at Table 4. Ra, Rb and Rc represent the signal data input. Figure 9 also shows 16 different memory location (R0 until R15) which operated by 16-bit instruction. In the instruction value, if the first number of the instruction given is 0 (for example 0xxx), the system will run the instruction ADD. Meanwhile if the first number of the instruction given is 1, the system will run the instruction SUB, executing the operations that have been set as in Table 4. The third (Ra) and fourth (Rb) (refer to Figure 8) number is the memory location of the source operand. Therefore, the second number (Rd) is memory location of the register destination after the two sources operand have perform the instruction.

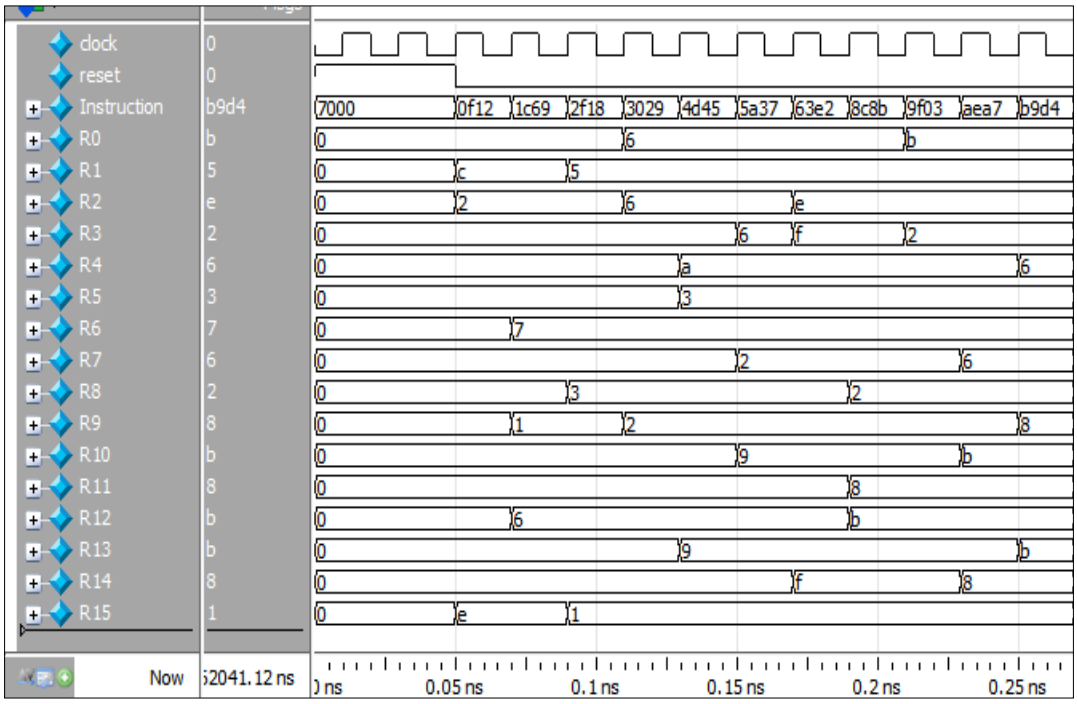

Figure 9: Simulation result for the proposed coprocessor

Figure 10 shows the simulation of performing the add operation with the 3 input signal, Rd = Ra+Rb by given 16-bit instruction for the memory location.

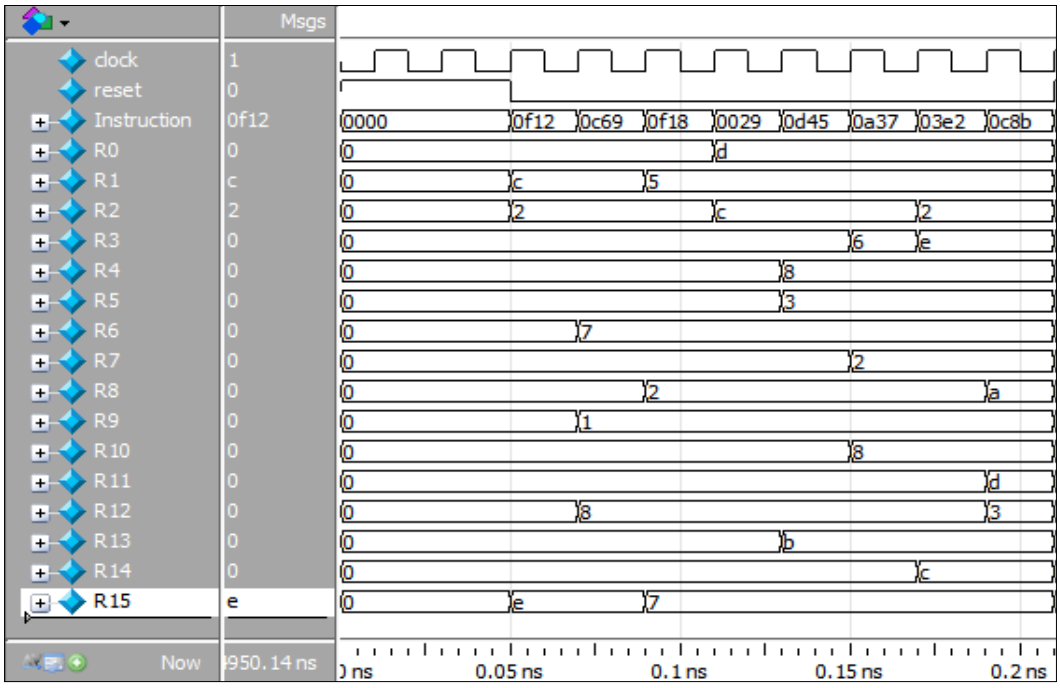

Figure 10: Simulation result of Instruction ADD

Figure 11 shows the simulation result, performing the subtraction operation with the 3 input signals, Rd = Ra-Rb.

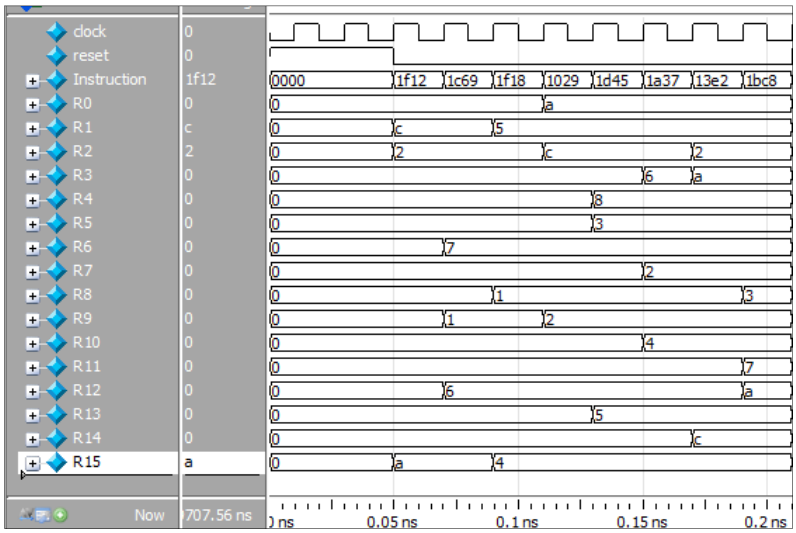

Figure 11: Simulation result of Instruction SUB

Figure 12 have used an AND gate to operate the multiplication rules, Rd = Ra & Rb.

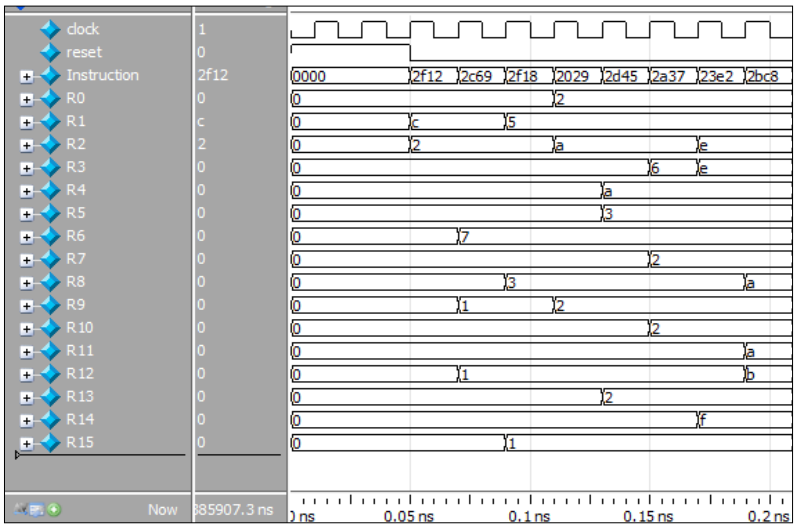

Figure 12: Simulation result of Instruction AND

Figure 13 have used an OR gate to operate the logical disjunction, Rd = Ra | Rb.

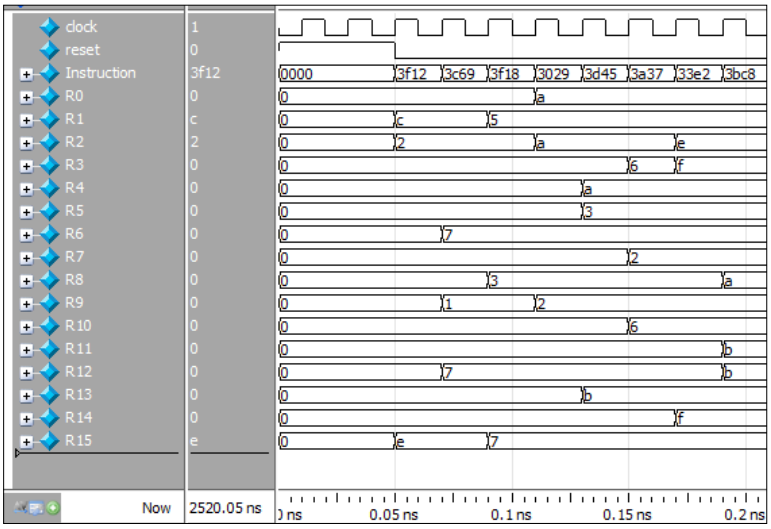

Figure 13: Simulation result of Instruction OR

Figure 14 shows the simulation result when performing operation of exclusive disjunction (XOR gate) with the 3 input signal, Rd = Ra ^ Rb.

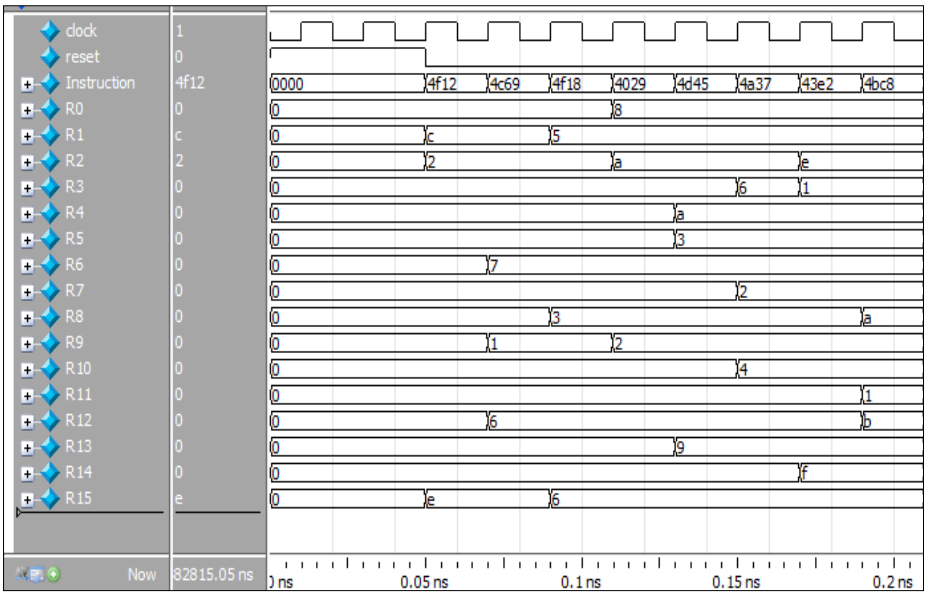

Figure 14: Simulation result of the Instruction XOR

Figure 15 shows simulation result when performing the inverting buffer or simply an inverter (NOT gate) which carry out operation with 2 input signal, Rd = $\overline{Ra}$.

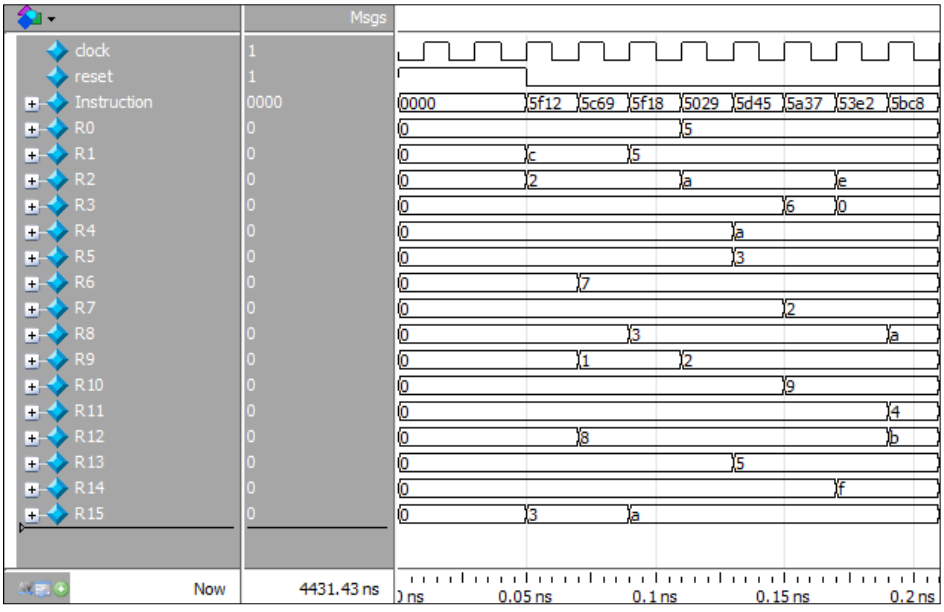

Figure 15: Simulation result of Instruction NOT

Result in Figure 16 shows the simulation result for operation of right rotating by 8 bits (ROR8) which carry out the operation Rd = Rb >> 2.

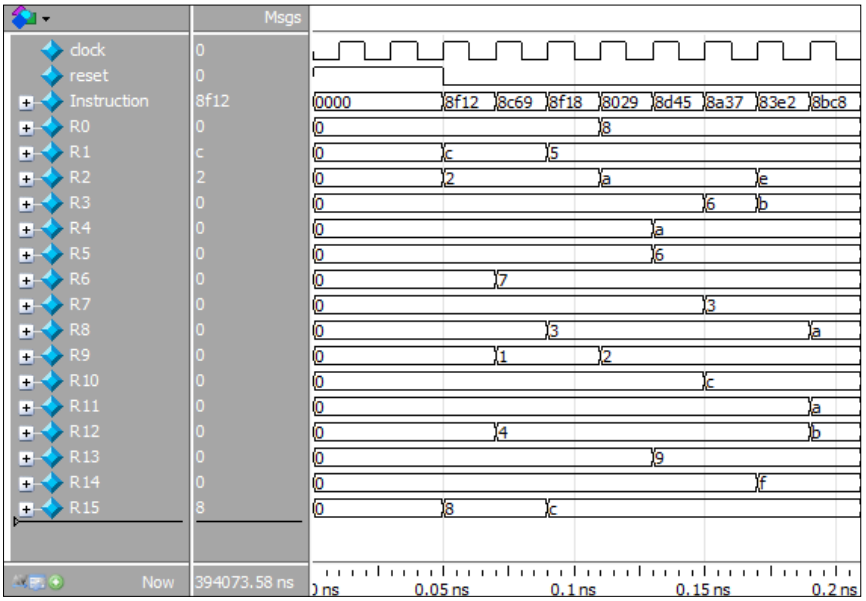

Figure 16: Simulation result of Instruction Rotate Right by 8 bit (ROR8)

Lastly, in Figure 17 shows the simulation result for a lookup table (LUT) operation and the value had set randomly to perform the hashing function.

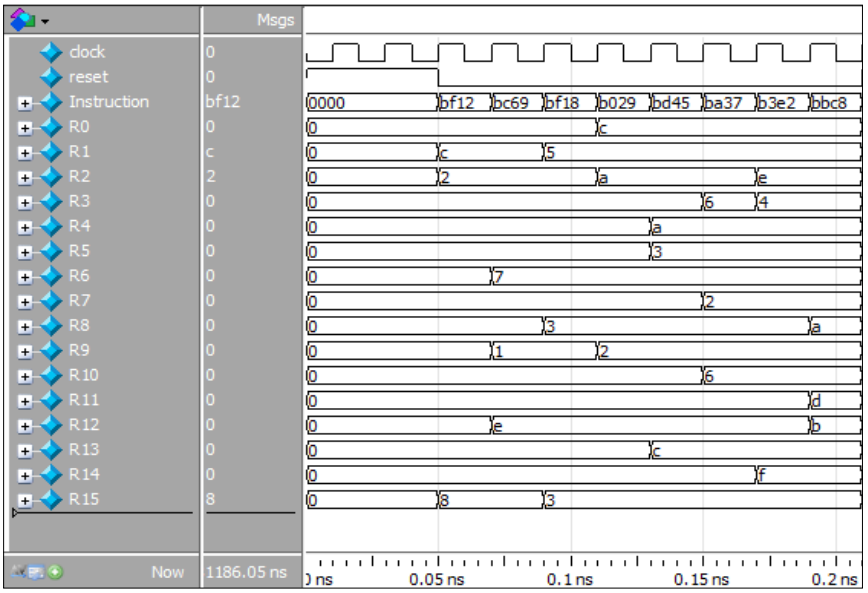

Figure 17: Simulation result of Instruction LUT

4. Conclusion

This study is the implementation of a cryptography coprocessor in FPGA by using the hash function. This cryptography coprocessor is able to help user computation environment to have crypto support which retains secrets and execute the software to prevent from the predictability logical or physical attack. Users can use this secure program as a basis for their secure applications such as financial transaction processing or high assurance digital signature generation. Cryptography coprocessor is widely use to offload the compute intensive cryptographic operations from the main

processor. Thus, this also will increasing the performance for the overall system. The cryptographic coprocessor design in Verilog Code is completely verified using Verilog HDL test bench. This study also performed coding of various blocks and verified the simulation waveform by using QuartusII Altera to run the program successfully. All results also have shown the concept of how the ISA work done with the 16 different memory location (R0 until R15) which operate by 16-bit instruction. Simulation results show that the proposed cryptographic coprocessor produces the correct outputs as in specification.

Acknowledgement

The authors would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] Cryptographic Application Scenarios (June 2011), by Vangelis Karatsiolis, Lucie Langer, Axel Schmidt, Erik Tews & Alexander Wiesmaier, Technische Universit¨at Darmstadt, Department of Computer Science, Hochschulstraße 10, D-64289 Darmstadt, Germany

[2] Cryptographic Algorithms for Secure Data Communication [accessed May 2011], Zirra Peter Buba & Gregory Maksha Wajiga

[3] A Review on Symmetric Key Encryption Techniques in Cryptography, by Qahtan M. Shallal and Mohammad Ubaidullah Bokhari, August 2016. International Journal of Computer Applications

[4] Computational Intelligence and Security, International Conference, CIS 2005, Xi'an, China, December 15-19, 2005, Proceedings, Part II

[5] Design and implementation of a private and public key crypto processor and its application to a security system, by IEEE Transactions on Consumer Electronics [Accessed in March 2004]

[6] RSA Public Key Cryptography Algorithm, July 2017, International Journal of Scientific & Technology Research 6(7):187-191